

REMARKS/ARGUMENTS

Applicants respectfully request reconsideration of this application.

All Applicants' Claims and cited art Ali et al ("Ali", WO 01/44932)

All of Applicants' claims stand rejected under Ali for either a 102(e) (all claims but claim 4), a 103(a) (claim 4), or 102(e) and 103(e) (claim 52).

Thus Applicants will discuss the distinction between Ali and Applicants' claim 1.

While Ali and Applicants invention are similar in many respects (XML, an eventual display), we submit that there is a fundamental difference in how the display information is generated.

Applicants' claim 1 recites:

1. A method comprising:
 - receiving a schema;
 - receiving an instance;
 - receiving a display specification; and
 - generating a display.

Ali is cited as anticipating this as noted at Office Action page 3 paragraph 3:

As per independent claim 1, Ali discloses a method comprising: receiving a schema (Figure 4A item 410); receiving an instance (Figure 4A item 414); receiving a display specification (Figure 4A item 412); and generating a display (Figure 4A item 424).

With all due respect, while Ali Fig. 4A shows groups of elements, a careful reading of the description of Fig. 4A does not support this interpretation. For example, Fig. 4A is

narrowly drawn to use a mouse control panel file 416, which contains the mouse control

panel UI settings and controls. Additionally, the Ali scheme sends the XSL output to the browser and the browser generates the UI. The whole approach is oriented to namespace navigation.

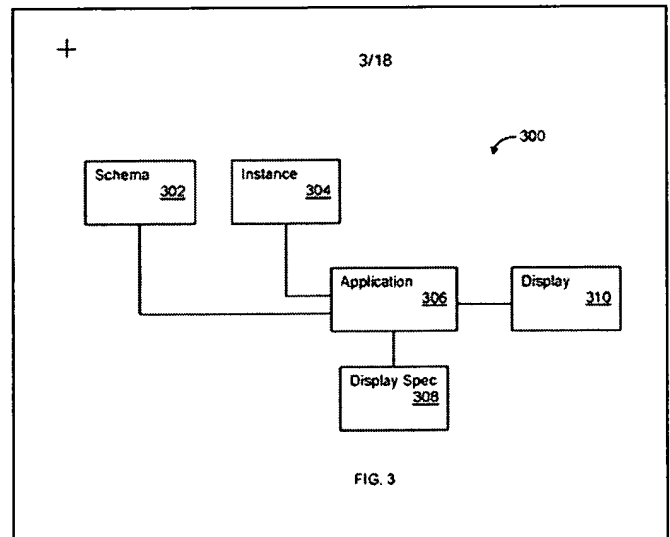
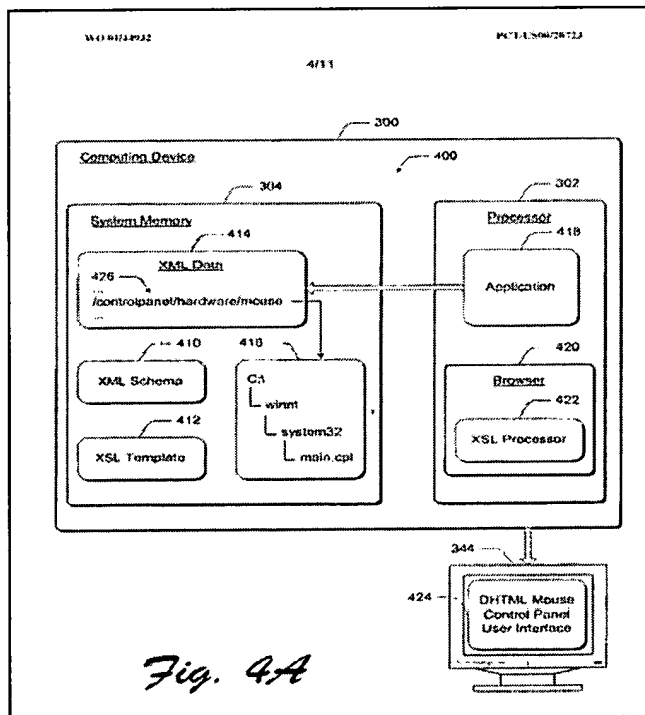
As the Examiner is aware the style sheet XSL is a template for an instance. The XML thus defines the UI. Thus a browser does not know where a data file is coming from as XSL does this. Thus the XSL cannot build the UI.

Additionally the presentation in Ali is display data.

In contrast, in Applicants' approach the presentation is of application data.

Thus a display specification influences how application data is displayed. Pattern matching is done to the display specification. The result is that Applicants' invention uses application (user) data, whereas Ali's XSL uses display data.

A comparison of Applicants' Fig 3 and Ali Fig 4A shows typological differences that do not allow a morphological equivalence.



Applicants' description at paragraphs [0044] through [0052] is a good example of the distinction which Applicants are trying to convey.

For the reasons stated above, Applicants submit that Ali does not anticipate Applicants' claim 1 and those claims dependent on it. Applicants submit that Ali does not anticipate independent claims 20, 38, 49, and 58 (or claims dependent on these) which also have a display specification.

With respect to claim 4 (which is dependent on independent claim 1) which stands rejected under 103(a) as obvious in light of Ali and Official Notice, Applicants submit that per the arguments above, Ali no longer anticipates Applicants' claim 1 and therefore the Official Notice standing alone does not make obvious Applicants' claim 4.

With respect to claim 52 (which is dependent on independent claim 49) which stands rejected under 103(a) as obvious in light of Ali and Official Notice, Applicants submit that per the arguments above, Ali no longer anticipates Applicants' claim 49 and therefore the Official Notice standing alone does not make obvious Applicants' claim 52.

For the above reasons Applicants submit that all claims are in condition for allowance and Applicants request such.

Example of distinction between Ali and Applicants' technique

Applicants submit that the examples and diagrams below provide an overview of the distinction between current approaches using XSLT processing (such as Ali) and Applicants' Display Specification processing.

Diagram 1 illustrates XSLT processing. An XML document and XLS stylesheet created a Destination document. The destination document is a translation and there is a one to one mapping between the stylesheet and the destination document. The destination document is then transferred to, for example, a browser which then does the UI generation.

Diagram 2 illustrates a Display specification processing of Applicants' invention. Here the XML document and a Display Spec are used to perform UI generation. The XML schema is queried for built-in output information.

As Diagram 1 and Diagram 2 illustrate, the UI generation is done at different places in the information flow and using different sources for information.

Diagram 3 illustrates the Process flow for XSLT processing. As illustrated, the stylesheet is read line by line and if a line does not contain a template then it is output as is, otherwise the template is used to get a result and the result is output.

Diagram 4 illustrates the Process flow for Applicants' UI generation. Here both the XML document and Schema are read. Then if the output can be rendered by the schema the UI component is generated, otherwise the display specification is queried for schema and XML nodes, and presentation rules using the schema and XML are used to generate the UI component.

Diagram 5 is a Schema that will be used to illustrate the differences in an actual example. Diagram 6 is an XML document. Diagram 7 illustrates a stylesheet and the resulting output of the Diagram 5 schema, and diagram 6 document using the diagram 7 stylesheet as rendered in browser.

Diagram 8 illustrates a Display specification. Diagram 9 illustrates a UI generated by the Diagram 5 schema, Diagram 6 XML document and the Diagram 8 display specification. Notice that tabbed presenters (Book, Book, Book, Book) are the result of the Diagram 8 line at 802. Also shown are a large edit box, a default edit box, and a checkbox that is check which is the result of the Diagram 5 line at 502.

Applicants submit that the differences in how the UI is generated are such that Ali does not anticipate Applicants' display specification approach.

XSLT Processing

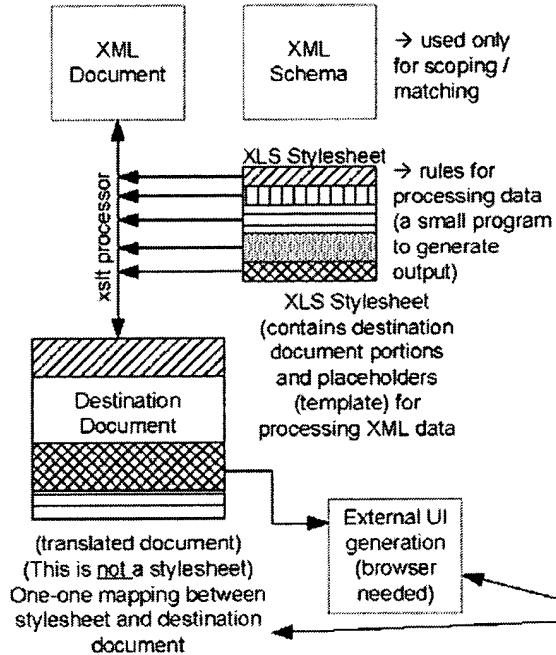


Diagram 1

Display Spec Processing

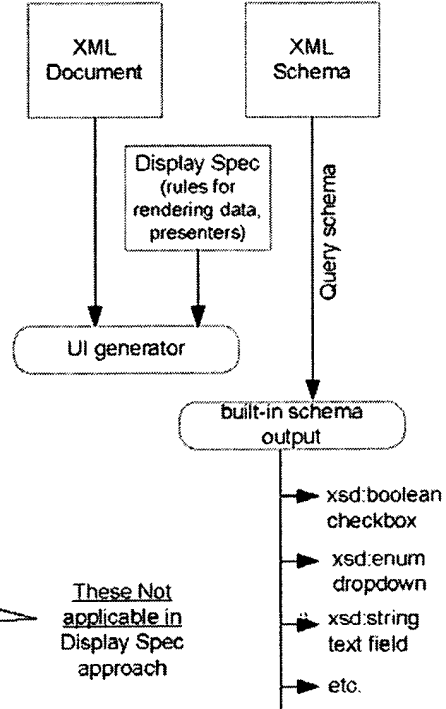


Diagram 2

Process Flow (XSLT Processing)

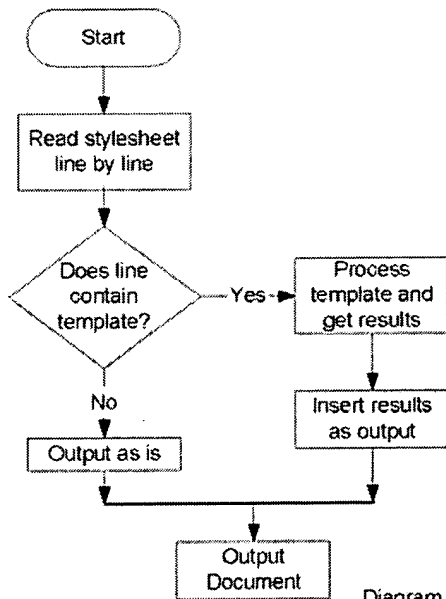


Diagram 3

Process Flow (UI generation)

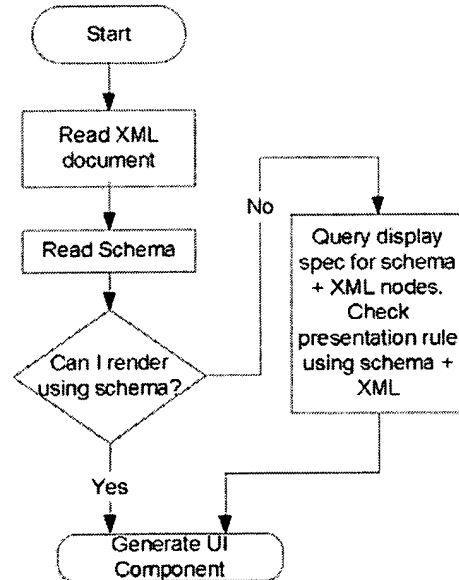


Diagram 4

SCHEMA

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  elementFormDefault="qualified"
  fd:Section="true"
  targetNamespace="http://www.fulldegree.com/App/Lib/Book.xsd"
  fd:Xptr="file://xmldata/fulldegree/App/Lib/Book.xsd"
  xmlns="http://www.fulldegree.com/App/Lib/Book.xsd"
  xmlns:fd="http://www.fulldegree.com/Schema/System/FullDegree.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="Books">
    <xsd:complexType>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="Book"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Author"/>
        <xsd:element ref="Title"/>
        <xsd:element ref="Price"/>
        <xsd:element ref="Paperback"/>
      </xsd:sequence>
      <xsd:attribute name="category" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Author" type="xsd:string"/>
  <xsd:element name="Title" type="xsd:string"/>
  <xsd:element name="Price" type="xsd:float"/>

502 → <xsd:element name="Paperback" type="xsd:boolean" default="true"/>

</xsd:schema>
```

Diagram 5

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<Books>

  <Book category="reference">
    <Author>Nigel Rees</Author>
    <Title>Sayings of the Century</Title>
    <Price>8.95</Price>
    <Paperback>true</Paperback>
  </Book>

  <Book category="fiction">
    <Author>Evelyn Waugh</Author>
    <Title>Sword of Honour</Title>
    <Price>12.99</Price>
    <Paperback>true</Paperback>
  </Book>

  <Book category="fiction">
    <Author>Herman Melville</Author>
    <Title>Moby Dick</Title>
    <Price>8.99</Price>
    <Paperback>false</Paperback>
  </Book>

  <Book category="fiction">
    <Author>J. R. R. Tolkien</Author>
    <Title>The Lord of the Rings</Title>
    <Price>22.99</Price>
    <Paperback>false</Paperback>
  </Book>

</Books>
```

Diagram 6



Stylesheet

Simplified Stylesheets

Example: A Simplified Stylesheet

This example shows a stylesheet that takes the form of an HTML skeleton page, with XSLT instructions embedded within it to pull data from the source document. The stylesheet is in the download file obtainable from <http://www.wrox.com>. It has the filename `simplified.xsl`, and can be used together with the data file `books.xml`.

The complete stylesheet is as follows:

```
<?xml-stylesheet href="http://www.w3.org/1999/XSL/Transform" type="text/xsl"?>
<xsl:stylesheet version="1.0">
  <html>
    <head>
      <title>A list of books</title>
    </head>
    <body>
      <h1>A list of books</h1>
      <table border="1">
        <xsl:for-each select="//book">
          <xsl:sort select="author"/>
          <tr>
            <td><xsl:value-of select="author"/></td>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="category"/></td>
            <td><xsl:value-of select="price"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
```

When you run this against the file `books.xml` (which is listed on page 70 in Chapter 2), the output is a sorted table showing the books, as follows:

Author	Title	Category	Price
Erlyn Waugh	Sword of Honour	Action	112.99
Herman Melville	Moby Dick	Action	98.99
J. R. R. Tolkien	The Lord of the Rings	Action	22.99
Nigel Rees	Sagas of the Century	Reference	3.95

Diagram 7



DISPLAY SPECIFICATION

```
<?xml version="1.0" encoding="UTF-8"?>
<DisplaySpecD
  displayNameNamespace="http://www.fulldegree.com/App/Lib/BookD.xml"
  fd:Name="BookD.xml"
  fd:Section="true"
  fd:Xptr="file:/xmldata/fulldegree/App/Lib/BookD.xml"
  xmlns="http://www.fulldegree.com/Schema/System/FullDegree.xsd"
  xmlns:any="http://www.fulldegree.com/Schema/System/ReservedAny.xsd"
  xmlns:fd="http://www.fulldegree.com/Schema/System/FullDegree.xsd"
  xmlns:bk="http://www.fulldegree.com/App/Lib/Book.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ElementD
    Edit="false"
    Name="bk:Books"
    displayNameNamespace=".">
    <Presenter
      Name="AllBooks"
      802 → ObjectName="com.fulldegree.display.presenters.TabbedTreePresenter"
      fd:BackgroundColor="204, 204, 204">
    </Presenter>
    </ElementD>

    <ElementD
      FontStyle="Bold"
      Name="any:Author">
      <Editor
        Name="labelWidget"/>
      </ElementD>

    <ElementD
      Name="any:Title">
      <Editor
        Name="textWidget"
        Width="large"/>
      </ElementD>

  </DisplaySpecD>
```

Diagram 8

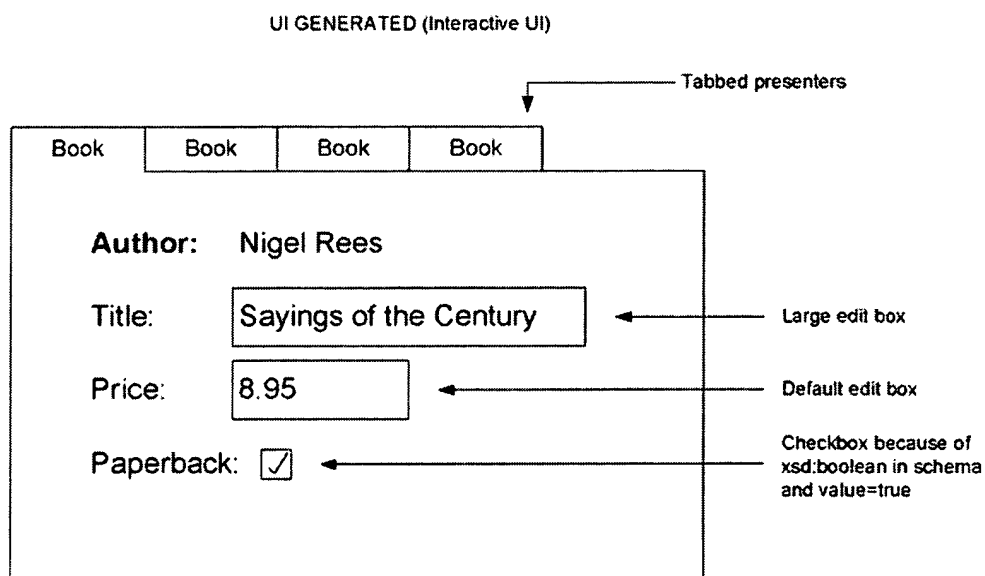


Diagram 9

CONCLUSION

Applicants submit that any claim not directly discussed is addressed via the independent claim discussion on which it is dependent.

Applicants respectfully submit that all claims are in condition for allowance, and request allowance of all claims.

The Examiner is invited to call Alan Heimlich at 408 253-3860 if there remains any issue with allowance. Cleartext email communication is authorized.

Respectfully submitted,

Heimlich Law

05/27/2005

Date

Customer No. 40418

5952 Dial Way
San Jose, CA 95129

Tel: 408 253-3860

Eml: alanheimlich@heimlichlaw.com



Digitally signed by Alan Heimlich
DN: CN = Alan Heimlich, C =
US, O = Heimlich Law

Alan Heimlich / Reg 48808

Attorney for Applicant(s)



The following has been received by the USPTO on the date stamped hereon.	
RECEIVED:	
Express Mail PO to Addressee: ED 510367756 US	Date Mailed: 05/27/2005
Application No. 10/033657	
Docket No. 005822.P001	

CONTENTS:	
PTO Transmittal Form (PTO/SB/21) – 1 pg	
Extension of Time Request (PTO/SB/22) – 1 pg	
Credit Card Payment Form (PTO-2038) - Total payment \$ 510.00 on CC ... 8366	
Response to Office Action of 11/29/2004 - 13 pgs	
This return receipt postcard – 1 postcard	
Certificate of Express Mail PO to Addressee with a copy of this return receipt postcard – 1 pg	

EXPRESS MAIL CERTIFICATE OF MAILING

Express Mail Post Office to Addressee No.: ED 510367756 US
Date of Deposit: 05/27/2005

I hereby certify that I am causing this correspondence, and the documents as indicated on the postcard representation above, to be deposited with sufficient postage to the United States Postal Service for Express Mail Post Office to Addressee service on the date of deposit indicated above, and addressed to the address below.

Mail Stop: AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Alan Heimlich

Printed Name of Person Performing Mailing

Signature

Digitally signed by Alan Heimlich
DN: CN = Alan Heimlich, C =
US, O = Heimlich Law

05/27/2005

Date